In the Claims:

Claims 1-9 (CANCELLED)

10. (CURRENTLY AMENDED) A method for generating interrupt commands for a microprocessor system, the method comprising:

storing interrupts in a pending interrupts register;

storing priority values associated with the stored interrupts in a plurality of priority registers coupled to the pending interrupts register;

loading a plurality of counters coupled in cascade to the plurality of priority registers with the stored priority values;

incrementing at predetermined intervals the priority values loaded in the plurality of counters;

comparing the incremented priority values for identifying the interrupt having a highest priority if an interrupt service routine is not being executed;

processing the interrupt having the highest priority by generating an interrupt command and an interrupt vector identifying the interrupt service routine to be executed; ~~and~~

canceling the interrupt having the highest priority from the pending interrupts register and its priority value from the plurality of priority registers; and

wherein the incrementing is based upon increment signals having different periods, each period corresponding to a particular interrupt that is associated with a corresponding counter and the increment signal of one of the plurality of

counters has a period shorter than a ratio between a maximum latency of one of the interrupts and a difference between maximum and minimum priority values.

11. (CANCELLED)

12. (CANCELLED)

13. (PREVIOUSLY PRESENTED) A method according to Claim 10, further comprising performing the following for managing nested interrupts:

storing the priority value of the interrupt being processed in a memory buffer;

comparing the priority values stored in the plurality of counters with the priority value stored in the memory buffer for identifying a new interrupt having the highest priority;

if the new interrupt has the highest priority, then stopping execution of the interrupt service routine and storing in a stack register the priority value corresponding to the interrupt service routine whose execution was stopped;

generating a new interrupt command and a new interrupt vector for identifying a new interrupt service routine to be executed based upon the new interrupt having the highest priority; and

canceling the priority value from the stack register corresponding to the interrupt service routine whose execution was recently serviced.

14. (CURRENTLY AMENDED) A method for generating interrupt commands for a microprocessor system, the method comprising:

storing interrupts in a pending interrupts register;

storing priority values associated with the stored interrupts in a plurality of priority registers coupled to the pending interrupts register;

loading a plurality of counters coupled in cascade to the plurality of priority registers with the stored priority values;

incrementing at predetermined intervals the priority values loaded in the plurality of counters;

comparing the incremented priority values for identifying the interrupt having a highest priority; ~~and~~

processing the interrupt having the highest priority; and

wherein the incrementing is based upon increment signals having different periods, each period corresponding to a particular interrupt that is associated with a corresponding counter, and the increment signal of one of the plurality of counters has a period shorter than a ratio between a maximum latency of one of the interrupts and a difference between maximum and minimum priority values.

15. (ORIGINAL) A method according to Claim 14, wherein processing the interrupt having the highest priority comprises generating an interrupt command and an interrupt vector identifying an interrupt service routine to be executed.

16. (ORIGINAL) A method according to Claim 14, further comprising canceling the interrupt having the highest priority from the pending interrupts register and its priority value from the plurality of priority registers after processing the interrupt having the highest priority.

17. (CANCELLED)

18. (CANCELLED)

19. (ORIGINAL) A method according to Claim 15, further comprising performing the following for managing nested interrupts:

storing the priority value of the interrupt being processed in a memory buffer;

comparing the priority values stored in the plurality of counters with the priority value stored in the memory buffer for identifying a new interrupt having the highest priority;

if the new interrupt has the highest priority, then stopping execution of the interrupt service routine and storing in a stack register the priority value corresponding to the interrupt service routine whose execution was stopped;

generating a new interrupt command and a new interrupt vector for identifying a new interrupt service routine to be executed based upon the new interrupt having the highest priority; and

canceling the priority value from the stack register corresponding to the interrupt service routine whose execution was stopped.

20. (CURRENTLY AMENDED) A control circuit for generating interrupt commands for a microprocessor system comprising:

a pending interrupts register for storing interrupts;

a plurality of priority registers coupled to said pending interrupts register for storing priority values associated with the stored interrupts;

a plurality of counters coupled in cascade to said plurality of priority registers and being loaded with the stored priority values, the stored priority values being incremented at predetermined intervals;

a priority comparing circuit coupled to said plurality of counters and to said pending interrupts register for comparing the incremented priority values for identifying the interrupt having a highest priority if an interrupt service routine is not being executed, and generating an interrupt request signal and an internal signal corresponding to the interrupt having the highest priority stored in said pending interrupts register; ~~and~~

a logic processing circuit coupled to said priority comparing circuit and receiving the interrupt request signal and the internal signal, and generating for the microprocessor system an interrupt command and an interrupt vector identifying an interrupt service routine to be executed; and

wherein the incrementing is based upon increment signals having different periods, each period corresponding to a particular interrupt that is associated with a corresponding counter and the increment signal of one of the plurality of

counters has a period shorter than a ratio between a maximum latency of one of the interrupts and a difference between maximum and minimum priority values.

21. (ORIGINAL) A control circuit according to Claim 20, wherein said plurality of counters receive increment signals derived from a system clock signal or from externally generated timing signals.

22. (ORIGINAL) A control circuit according to Claim 21, wherein the increment signals have respective periods that are a multiple of a period of the system clock signal.

23. (ORIGINAL) A control circuit according to Claim 22, wherein the increment signals have different periods.

24. (ORIGINAL) A control circuit according to Claim 20, wherein the interrupts are nested, and wherein said logic processing circuit comprises:

a memory buffer coupled to said priority comparing circuit for storing the priority value of the interrupt being represented by the internal signal; and

a stack register coupled to said memory buffer for storing the priority value corresponding to the interrupt service routine whose execution was stopped because of a new interrupt having a higher priority value.

25. (CURRENTLY AMENDED) A microprocessor system comprising:

a microprocessor; and

a control circuit for generating interrupt commands for said microprocessor, said control circuit comprising

a pending interrupts register for storing interrupts,

a plurality of priority registers coupled to said pending interrupts register for storing priority values associated with the stored interrupts,

a plurality of counters coupled in cascade to said plurality of priority registers and being loaded with the stored priority values, the stored priority values being incremented at predetermined intervals,

a priority comparing circuit coupled to said plurality of counters and to said pending interrupts register for comparing the incremented priority values for identifying the interrupt having a highest priority, ~~and~~

a logic processing circuit coupled to said priority comparing circuit and generating for said microprocessor an interrupt command and an interrupt vector identifying an interrupt service routine to be executed, and

wherein the incrementing is based upon increment signals having different periods, each period corresponding to a particular interrupt that is associated with a corresponding counter and the

increment signal of one of the plurality of counters has a period shorter than a ratio between a maximum latency of one of the interrupts and a difference between maximum and minimum priority values.

26. (ORIGINAL) A microprocessor system according to Claim 25, wherein said priority comparing circuit generates an interrupt request signal and an internal signal representing the interrupt having the highest priority stored in said pending interrupts register; and wherein said logic processing circuit receives the interrupt request signal and the internal signal for generating the interrupt command and the interrupt vector identifying the interrupt service routine to be executed.

27. (ORIGINAL) A microprocessor system according to Claim 25, wherein said plurality of counters receives increment signals derived from a system clock signal or from externally generated timing signals.

28. (ORIGINAL) A microprocessor system according to Claim 27, wherein the increment signals have respective periods that are a multiple of a period of the system clock signal.

29. (ORIGINAL) A microprocessor system according to Claim 28, wherein the increment signals have different periods.

30. (ORIGINAL) A microprocessor system according to Claim 25, wherein the interrupts are nested, and wherein said logic processing circuit comprises:

a memory buffer coupled to said priority comparing circuit for storing the priority value of the interrupt corresponding to the internal signal; and

a stack register coupled to said memory buffer for storing the priority value corresponding to the interrupt service routine whose execution was stopped because of a new interrupt having a higher priority value.